PCT/IS00/00010.

#2

10/089237

REC'D 16 OCT 2000

WIPO PCT

IS00/10

4

Einkaleyfastofan REYKJAVÍK

LÝÐVELDIÐ ÍSLAND

Hér með staðfestist að meðfylgjandi eru rétt afrit af gögnum sem upphaflega voru lögð inn hjá Einkaleyfastofunni vegna neðargreindrar einkaleyfisumsóknar.

This is to certify that the annexed is a true copy of the documents as originally filed with the Icelandic Patent Office in connection with the following patent application.

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

(21) *Umsóknarnúmer*
Patent application number — **5207**

(71) *Umsækjandi*
Applicant(s) — **Marel hf.**

(22) *Umsóknardags.*
Date of filing.. — **1.10.1999**

ELS
EINKALEYFASTOFAN

Reykjavík, 4.október 2000

Sigurlin Bjarney Gisladottir
Sigurlin Bjarney Gisladottir

MULTI-FILTER

The present invention relates to a method for weighing items and in particular to a method of filtering signals representing the weight of an item.

The invention is applicable in all measuring instruments which indicate a DC value calculated from a signal which also has unwanted AC components of one or more unknown frequencies.

This is, for example, the case in voltmeters, ampere meters and weighing-scales. Normally, the indicated result is the mean value of the signal measured for a much longer time than the expected period of the slowest AC component. This method is well known and reliable, but has the drawback of being very slow in situations where measuring speed is of importance.

The invention is of particular importance in high speed weighing scales. In that case, the DC value is the result (the weight) but because of vibrations in the mechanics of the scale and its base, a lot of unwanted AC components are added to the DC value.

BACKGRUOND FOR THE INVENTION AND INTRODUCTION TO THE INVENTION.

Electronic scales comprising a load cell providing readout in terms of voltage representing the weight of the item today perform weighing of items. This readout is typically passed through an analogue to digital (A/D) converter that converts the electrical signal to a digital signal normally represented by a bit representation of a number. This bit representation is then manipulated further in order to provide a read out of the weight of the item on display.

When an item is to be weighed by such an electronic scale - or in general by an ordinary scale - the item is arranged on a scale platform. As the mechanical parts of the scale is not weightless the scale platform and the parts connected to the platform may be able to vibrate which in turn influences the read out - the measurement - of the scale. The vibration may in general be induced by at least two measures: the way the item is arranged on the scale platform and the vibrations of the surroundings.

07 01.10.99

The first measure may for instance be instanced by dropping an item on to the scale platform - a situation, which for instance occur in a process line in which items are being conveyed from a processing station past a weighing station to a grading and/or packing station.

The second measure may for instance be instanced when the scale is applied in an environment in which heavy machinery inducing vibrations is present.

In both cases, or of course in combinations of these, the signal of from the load cell and in turn the readout of the weight will follow a curve which ideally can be described as the motion of a damped oscillation, where the mean value of curve is the final steady state value, i.e. the weight which is to be determined.
In weighing scales there are mostly 3 types of unwanted AC components. These occur due to:

1 Steady vibrations in the base of the scale from nearby machinery.

2 Vibrations which occur when an object is placed on the platform. These vibrations cease when the mechanism becomes stable.

3 Short-duration mechanical transients which hit the base of the scale.

Every scale designer wants to make a fast scale, which displays the correct result as soon as an object is placed on the platform. A faster scale saves both time and money. At the same time, he wants the scale to be insensitive to vibrations and shocks that strike the platform or the base of the scale. Unfortunately, it is difficult to achieve both features at the same time using only one filter.

If a fast filter is used, the scale will be fast, but the corner frequency of the filter is high and AC components inside the passband are not filtered, leading to unsteady results. Conversely, if a slow filter is used (low corner frequency), the scale will be less sensitive to transients and AC components, but the scale will be slow to display results.

Weighing scales must show results, along with an indication on the stability of the scale. As a result, designers usually use a slow filter to minimise the risk of the scale being unsteady all the time, and therefore being of no use.

The invention is very useful, as it allows the use of fast filters without the risk of the scale being useless if the base is vibrating.

In known systems the signal from the load cell is filtered either digitally or analogue with a cut off filter removing all parts of the signal having a higher frequency than the cut-off frequency. By this cutting off a steady readout is obtained faster than if no filter is applied as the mechanical system then has to be in rest before a steady read out is available.

The cut-off frequency is typically a function of the weight of the moving parts of the scale, the weight of the item to weight, the mechanical damping characteristics of the scale and the vibrations induced by the surroundings. This means that the scale typically is equipped with more than one cut-off filter in order to be able to work fast in different environments and with items of varying sizes.

A major problem with these known system is however that if the variation of the weight of the items varies fast, for instance from one item to another, and/or the vibrations induced by the surroundings also varies fast, then different filters has to be selected often in order to have a response of the weight always being fast.

Especially the speed of the weighing has dramatically influence on the speed of for instance a grading process, as the weighing today normally is the operation limiting the speed of the grading process.

Until now, the change of cut-off filter is made manually as many attempts to automatically changing filter or selecting the optimum filter have been unsuccessfully. Such a manually change or selection of filter is in nature quite slow as it involves an operator changing the set-up of the scale.

Thus it an aim of the present invention to provide a method for automatically selecting a cut-off filter having the best cut-off frequency such weighing of items may be performed as fast as possible with a particular scale.

BRIEF DESCTIPTION OF THE INVENTION

This problem has been solved by means of the present invention providing a method for processing a stream of digital signals utilising at least two processing means adapted to processing digital signals; each processing means processes by varying degrees, which method comprising:

- providing one intermediate output signal for each processing means applied; the intermediate output signals are based on processing a set or sets of digital signals selected from the stream of digital signal,
- assigning an intermediate output to an output signal if the intermediate output signal in question is
  - the one processed least and
  - if a stability requirement corresponding to the intermediate output signal in question have been fulfilled

In a broad aspect, the present method according to the present invention is applicable to any kind of processing of a stream of digital signal being the result of for instance an analogue-to-digital conversion, the analogue signal being the result of some recognition - or measuring - of one or more physical quantity, such as a weight of an item.

In the present content, a digital signal denotes a signal being represented by a numerical value, and a stream of digital signals denotes a series of digital signals being ordered sequentially by time. In a preferred embodiment of the present invention the stream of digital signals is a substantially constant inflow of digital signals to the processing means but the stream of digital signal may also preferably be a set of data representing digital signals ordered sequentially in time.

The processing means are preferably numerical algorithms executed in a digital processor such as a microcomputer, but hardware processing means may also be used in connection with the present invention.

The processing means, which in general are different, are different in the sense that the processing performed by them are said to vary in degree. By this varying processing degree is meant that the for instance the cut-off frequencies of the processing means, in case these are characterised as filters are different. In other cases, the difference may be the way they process the digital signals such as one processing giving a mean value and another processing means giving a filtered value (please note that in some case process providing a mean value is referred to a filtering process).

In all cases, the processing means processing least is the one taking away least information from the stream of digital signals compared with the processing means which takes away most information from the stream of digital signals.

In the broad aspect of the present invention each of the processing means provides an intermediate output signal. These signals are in general different, as they are the results of applying the processing means, which are different, to the stream of digital signals.

After these intermediate signals are made available by the processing means the method according to the present invention detects the signal being the most appropriate for the time being, i.e. the signal which is processed least and which fulfils a stability requirement. The most appropriate signal is assigned to be the output signal.

In a preferred embodiment of the present invention one set of digital signals is selected for each processing means applied so that an intermediate output signal provided by a processing means applied is based on a set of digital signals selected particular for the processing means in question.

In a general aspect of the present invention the stream of digital signal is arriving to the processing means successively/sequentially. The selection of the sets to be processed is preferably done so that signals to be processed is held in a list of signals whereof some may have been processed and whereof one or more of these signals is/are replaced when a new signal, a signal not yet processed, is available from means providing the signals.

In a preferred embodiment of the present invention, the method makes use of a first set of digital signal selected for the processing means processing most. Out of this first set of digital signals sub-set(s) is (are) selected to be processed by the remaining processing means. For instance one sub-set of digital signals may be every second digital signal comprises in the first set of digital signals.

BRIEF DESCRIPTION OF THE DRAWINGS

In the following the invention and particular preferred embodiments thereof will be described in greater with reference to the accompanying figures in which

Fig. 1 is a schematically drawing of an electronic weight,

Fig. 1 is a schematically drawing a scale according to the present invention,

Fig. 2 is a schematically drawing of the main electronic components of a scale according to the present invention,

Fig. 3 is a schematically drawing of a preferred embodiment of the filtering method according to the present invention,

Fig. 4 is a flow chart of a preferred embodiment of the filtering method according to the present invention,

Fig. 5. is a print-out of a program performing when launched a preferred embodiment of the filtering method according to the present invention

In a general aspect of the present invention the stream of digital signal is arriving to the processing means successively/sequentially. The selection of the sets to be processed is preferably done so that signals to be processed is held in a list of signals whereof some may have been processed and whereof one or more of these signals is/are replaced when a new signal, a signal not yet processed, is available from means providing the signals.

In a preferred embodiment of the present invention, the method makes use of a first set of digital signal selected for the processing means processing most. Out of this first set of digital signals sub-set(s) is (are) selected to be processed by the remaining processing means. For instance one sub-set of digital signals may be every second digital signal comprises in the first set of digital signals.

BRIEF DESCRIPTION OF THE DRAWINGS

In the following the invention and particular preferred embodiments thereof will be described in greater with reference to the accompanying figures in which

Fig. 1 is a schematically drawing of an electronic weight,

Fig. 2 is a schematically drawing of the main electronic components of a scale according to the present invention,

Fig. 3 is a schematically drawing of a preferred embodiment of the filtering method according to the present invention,

Fig. 4 is a flow chart of a preferred embodiment of the filtering method according to the present invention,

Fig. 5. is a print-out of a program performing when launched a preferred embodiment of the filtering method according to the present invention

07 01.10.99

GENERAL DESCRIPTION OF THE INVENTION

In general the invention relates to the problem of predicting the stable condition of a mass oscillating in a damped motion. Considering a mass oscillating in a damped system one way to describe the motion of the mass oscillating is by keeping track of its position during time. This may be described by for instance $x = f(t)$ where the function f in general is unknown. It has been shown that the motion of a mass oscillating in a damped linear system may be described by a superposition of elementary motions where each of these elementary motions may be described by functions depending on cosines or sinus, for instance:

$$x(t) = a_0 + \sum a_i(t)\cos(f^*t), \quad (1)$$

where a_0 typically is a constant, $a_i(t)$ are amplitude functions going to zero with increasing time, t, and f is a frequency. This indicate that x(t) goes towards a_0 with increasing time, and the rest position of the mass performing damping oscillation is actually a_0.

In relation to the invention the problem to be solved in connection with scales is to estimate the rest position, in mathematical terms a_0, before the mass actual comes to a rest, as this often takes very long time a time which is not available in modern grading system etc.

One could actually try to fit equation (1) to a series of signal, but such a fit requires estimation of the unknown terms and attempts to figure out the actual expression for $a_i(t)$. Such a procedure would require extensive use of a computer which again imply a too long processing time.

Instead the damped oscillation of the motion may is studied in the frequency domain, i.e. a representation of the amplitude, expressed in term of (dB), of the motion as a function of frequencies.

DETAILIED DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

In fig.1 a typical electronic scale 10 is shown. The scale 10 comprises a base part 20 wherein a load cell 60 is situated, a scale platform 40 on which the item to be weight is placed and display 50. The scale 10 also comprises a logical unit such as a computer or a micro controller for transforming the signal coming from the load cell 30 to a read out on the display 50.

In fig. 2 the electronics comprised in the scale 10 are schematically depicted. The actual choice of electronic components is not crucial for the invention as ordinary known electronically components may be used. Referring to fig. 2 the load cell 60 which may comprise a strain gauge acting as a resistor in a Wheatstone bridge circuitry is actuated by for instance a rod connected to the scale platform 40.

The load cell 60 will thereby provide an electrical signal with a magnitude representing the force of the scale platform 40 applied to the load cell 60. This force will in general be different from the weight of the moving part of the scale plus the item times the gravity until these parts are in rest, as the movement of the parts involves acceleration of the parts.

The signal coming from the load cell 60 is then directed to a A/D-converter 70 from which a bit pattern representing the actual amplitude of the signal from the load cell is provided at a predetermined sampling rate. The sampling rate and the resolution of the A/D-converter may be determined by the physics of the scale.

The bit pattern is then directed to a micro-controller 80 being able to perform a filtering of the signal. Referring to Fig. 3, the filtering is performed by use of, for instance, five filters having the following cut-off frequencies:

Filter A = 5 H_z	(fastest weighing - more sensitive to exterior vibrations)
Filter B = 4 H_z	(fast weighing)
Filter C = 2 H_z	(normal weighing)
Filter D = 1 H_z	(slow weighing)
Filter E =0.5 H_z	(slowest weighing)

The filtering is performed by the following scheme. Each time a value is available at the output of the A/D-converter, this value is directed to the filtering means. The filtering means holds a list of values corresponding to a pre-selected number of earlier measured values. Each filter is activated and determines filtered intermediate filtered values, one for each filter, based on the new value and the old values except the oldest.

For each of the filters applied, the intermediate filtered values are checked for stability, i.e. the most recent determined intermediate value corresponding to a specific filter is compared with the value determined last time the same filter was activated. If the difference between these two successive intermediate values is within a certain limit then the signal filtered with the specific filter is said to be stable.

In many practical applications more than one intermediate value is found to be stable and in that case the intermediate value corresponding to the filter filtering least is selected as the output value from the filtering routine. The reasoning behind choosing the value corresponding to the filter filtering least is that this choice will provide the earliest stable readout of the weight which readout reflects the final steady state of the weight.

Description of the filtering routine

According to the general aspect of the present invention a selection of the most appropriate filter is done. Before going into a detailed description of the selection routine and a preferred embodiment of this and the filters applied, the general structure of the filtering routine is explained with reference to Fig. 3.

The digital weight signal, Input value of Fig. 3, provided by the A/D-converter is filtered with a number of different filters, Filter A, Filter B, Filter C, Filter D and Filter E of Fig. 3. The output from each filter is checked for stability and the least filtered (fastest) steady value is selected. This will give the fastest possible response. Figure 4 is showing the Auto Response System principle.

As indicated in Fig. 3 the filters are providing results after different instances in time, for instance Filter A provides a result after approximately 0.3 seconds in a preferred embodiment of the invention.

In a preferred embodiment of the filter algorithm to be described below the following response times are achieved:

Filter A	result available after 0.3-0.5 seconds	(fastest response)
Filter B	result available after 0.6-0.8 seconds	(fast response)
Filter C	result available after 0.9-1.2 seconds	(normal response)
Filter D	result available after 1.3-2.0 seconds	(slow response)
Filter E	result available after 2.0- seconds	(slowest response)

These response times are features of the filter routine, which is not to be eliminated as the Filter E for instance has to work on many more values than for instance the Filter A.

The selection procedure is depicted in Fig. 4, which shows route diagram for the filter routine. The routine starts by receiving a value, in_val, from the A/D-converter. Then a threshold value is determined as the minimum resolution of the weighing scale, i.e. the bit resolution provided by the A/D-converter.

Based on the value received from the A/D-converter, the signal filtered by applying the four different filters shown in Fig. 4 (J4_filter, J3_filter, J2_filter and J1_filter).

The stability of the filtered signals is tested by a statement:

(fabs(old_val-new_val) < threshold/n)

wherein n takes the values of the number of items on which the filtering is based.

As shown in Fig. 4 the stability of the filters are tested successively starting with the filter having the highest cut-off frequency and once stability has been detected the output of the filtering routine is assigned the filtered value. After a filtered value has been determined the filtering routine is prepared for a next evaluation by assigning the old_val the newly

determined filtered values, new_val, for all the filters applied and the selected filtered value is returned from the routine to the electronics applied for displaying the value.

In some case no stability is detected and a filter is therefore not selected. In this case value provided by the A/D-converter is assigned to the filtered value and this value is then returned as if it was a value provided from one of the filters.

<u>Introduction to the filter algorithm</u>

In the following a preferred embodiment of the filter algorithm according to the present invention will be introduced. The features of the method according to the present invention may in general be described by the following item list:

- There are several low-pass filters running at the same time.
- The invention deals with the selection of the filters in real time
- The lowpass filters have different corner frequencies, but their exact implementation is not important.
- The software selects in real time the fastest steady filter.
- If no filter gives a steady result the fastest one is selected.

The method according to the present invention works in the following way, when considering different situations:

<u>Steady vibrations attack the base of the scale.</u>

- Only the slowest filters will produce a steady result, and the result from the one with the highest corner frequency will be used.
- The scale will work accurately, but will be slow.

<u>Vibrations occur when a object is placed on the platform and stop when the mechanic becomes stable</u>

- A DC signal will be formed when the object it put on the platform. An AC signal will be added because of the vibration.
- The results from all filters are monitored and the result from the first filter to show a steady result will be used.

- Which filter will be first to give a steady result depends on the system character and the vibrations,
- In a well-designed system, the fastest filter would be the first to become stable if no vibration exists in the base.

<u>Transient shocks on the base</u>

- In this case the fastest filters will give unsteady result, and the software will look for a slower filter not affected by the shock.
- The displayed result remains steady until the slowest filter gives an unsteady result. This gives the scale the stability of the slowest filter.

Steady calculation.

One way of detecting a steady result from a filter is to calculate the difference between the last two readings and compare it to a predetermined value. If the difference is greater than the value, the filter is unstable. The time between any two readings is constant, for example 100 milliseconds.

It is also possible to use a change in the final displayed result (weight) as a indication of a steady result. This is an advantage in weighing scales where the displayed weight must be rounded in certain way to comply with "Weights and Measures " regulations

This method gives better results but requires more processing power.

In the following the method according to the present invention is described with reference to a program which when launched in a computer will perform the processing (filtering) of the signals. The program is written in Pascal and comprises the following statements:

*	Filter05	A variable that holds the result of a slow 0.5 Hz filter
*	Filter1	A variable that holds the result of a 1 Hz filter
*	Filter2	A variable that holds the result of a 2 Hz filter
*	Filter3	A variable that holds the result of a 3 Hz filter
*	Filter5	A variable that holds the result of a fast 5 Hz filter.
*	Steady05	A flag that is true if Filter05 gives a steady result
*	Steady1	A flag that is true if Filter1gives a steady result
*	Steady2	A flag that is true if Filter2 gives a steady result
*	Steady3	A flag that is true if Filter3 gives a steady result
*	Steady5	A flag that is true if Filter5 gives a steady result

```
*      Result         A variable that will be displayed after rounding.
*
Repeat
       UpdateAllFilters ;                        * All filters are updated
       Display  :=   Filter5 ;                   * Use fastest filter if no filter is stable
       If Steady05   THEN result:= Filter05 ;
       If Steady1    THEN result:= Filter1 ;
       If Steady2    THEN result:= Filter2 ;     * Display gets the value of
       If Steady3    THEN result:= Filter3 ;     *  the fastest steady filter
       If Steady5    THEN result:= Filter5 ;
       DisplayAndRound(result) ;                 * Process and display the result
until false
```

<u>Description of the preferred embodiment of the filter algorithm</u>

In the following the filter algorithm is described in details. Firstly, the basic algorithm is described exemplified and secondly an actual preferred implementation of the algorithm is described. The implementation is depicted in the accompanying Fig. 5.

In general, a value (termed in_val) reflecting the signal from the A/D value arrives directly from the A/D converter approximately 10 times pr. sec. The A/D converter has a internal filter so the response time direct from the A/D converter is 0.3-0.5 sec.

In the presently most preferred embodiment of the present application the filter algorithm is implemented in form similar to the one outlined below (only one of the filters is referenced as the other filters are quite similar):

```
1) float J_filter(float VALUE)
2) J_sum          =J_sum+value - J[J_pos]
3) J_array[J_pos]=VALUE
4) J_pos          =(J_pos+1)&(J_size-1)
5) return J_sum / J_SIZE;
```

The array J holds the values originating from the measurement conducted by the A/D-converter.

For a better understanding of the algorithm the following sequence of measured values are considered:

Number	1	2	3	4	5	6
Value	1000	990	1010	1005	950	1040

The J array holds the last four measured values, f.i.:

Array Pos	0	1	2	3
Value	1000	990	1010	1005

and J_pos is pointing at the oldest measured, f.i. J_pos = 0.

Now a new measured value "arrives", which in the present example is considered to be item number 5, and the value of this item is transferred to the filter by use of VALUE. The sum of the three old elements and the new one is calculated by statement 2). This new measured value is then stored in the J array at position O by statement 3). In order to prepare the filter for the next evaluation of a mean value J_pos is shifted to the position of the second oldest item - J_pos is counting 0, 1, 2, 3, 0, 1, 2, 3 ..... - by statement 4). By statement 5) the mean value of the four most recent measured values are calculated by evaluating the sum of these values divided by the total number of elements considered.

Now the J array holds of the following elements:

Array Pos	0	1	2	3
Value	**950**	990	1010	1005

The emphasised item (950) is the most recent measured element.

Now the number 6 item arrives, 1040, which also is transferred to the filter by use of VALUE. A new sum of the now for most recent values is again calculated and VALUE is entered into the J array at position number 1 - J_pos is referencing position number 1 - and J_pos is re-counted for next time application of the filter. The mean value is calculated and returned by use of VALUE.

Now the J array consist of the following elements:

Array Pos	0	1	2	3
Value	950	**1040**	1010	1005

The emphasised item (1040) is the most recent measured element.
These steps are applied each time a new value measure by the A/D-converter is recorded and the filter thereby provides a mowing average of the measured data.

According to the general aspect of the present invention more than one filter is applied. In this case - following the discussion which was put forward in the section general description of the invention - the total filter algorithm is operating on different level of representation of the raw-data. For instance if two filters is applied on a series of data comprising 8 elements then one filter is operating on all eight elements whereas the next filter is operating on every second element, i.e. element number 1,3,5,7.

If one choose to store all the measured values, then one has to use an amount of memory being equal to at least the maximum number of elements to be considered. As the available amount of memory in weighing systems often are limited the filter algorithm has been implemented in cascade. The cascade has the following structure:

J4_VAL=J4_filter(in_val)
J3_VAL=J3_filter(J4_val)

.

.

Jn_VAL=Jn_filter(Jn-1_val)

As it is shown the filters are called successively and the input to the next filter is the output from the newly finished filter. In this embodiment a lot of memory is saved as the input to the succeeding filters already are mean values based on the values on a lower level.

In the presently most preferred embodiment of the invention four filters are applied which in the cascade implementation requires 16+8+4+2=120 bytes of memory (each value requires 4 bytes of memory). In a non-cascaded implementation the same four filters would require 1024+64+8+2=4392 bytes of memory.

Furthermore, as the input to the next filter already is a mean value the speed of the filtering is increased as the mean value of the four elements shown in the table above may be determined as:

$$n_{1,2,3,4}/4 = n_{1,3}/2 + n_{2,4}/2$$

($n_{1,3}$ for instance refers to the sum of item 1 and 3). In the first filter, for instance, $n_{1,3}/2$ is evaluated and the evaluated value is passed on to the next filter evaluating $n_{1,3}/2$ + the input, whereby $n_{1,2,3,4}/4$ is evaluated.

CLAIMS

1. A method for processing a stream of digital signals utilising at least two processing means adapted to processing digital signals; each processing means processes by varying degrees, which method comprising:

- providing one intermediate output signal for each processing means applied; the intermediate output signals are based on processing a set or sets of digital signals selected from the stream of digital signal,
- assigning an intermediate output to an output signal if the intermediate output signal in question is
  - the one processed least and
  - if a stability requirement corresponding to the intermediate output signal in question have been fulfilled.

2. A method according to claim 1,wherein one set of digital signals is selected for each processing means applied so that an intermediate output signal provided by a processing means applied is based on a set of digital signals selected particular for the processing means in question.

3. A method according to claim 2, wherein a first set of digital signal is selected for the processing means processing most and the sets processed by the remaining processing means is/are sub-sets of the first set of digital signals.

4. A method according to any of the preceding claims, wherein the digital signals is representing responses to weighing of an item on a weighing scale, and wherein the digital signal is being provided successively/sequentially by means responsive to weighing of an item.

5. A method according to any of the preceding claims, wherein the processing of the digital signals is carried out substantially each time a digital signal is provided by the means responsive to the weighing of the item.

6. A method according to any of the preceding claims, wherein the processing means comprise(s) digital filters for filtering digital signal

7. A method according to claim 6, wherein the digital filters comprise low-pass filters.

8. A method according to claim 7, wherein the low-pass filters are running averaging algorithms numerically evaluating the mean values of the digital signals.

9. A method according to claim 8, wherein the sets of digital signal to be used in the running averaging algorithms are series of numbers 2, 4, 8, 16 numbers of digital signals.

10. A method according to claim 8 or 9, wherein the running averaging algorithms are applied successively; the next algorithm applied is applied based at least partly on the result of the previously applied algorithm.

11. A method according to any of the preceding claims, wherein the stability requirement to be fulfilled comprises a requirement to a maximum difference between two successive intermediate output values.

12. A method according to any of the claims 4-11, wherein the means responsive to the weighing of the item comprise(s) an A/D-converter providing an analogue signal representing a response from a weighing scale.

13. A method according to claim 12, wherein the digital signal is representing responses to measurement of an electrical quantity such as voltage, ampere or the like, and wherein the digital signal is being provided successively by means responsive to measuring the electrical quantity.

14. A method according to any of the preceding claims, wherein the intermediate signal being processed least is assigned to the output signal in case no intermediate output signal fulfils the stability requirement.

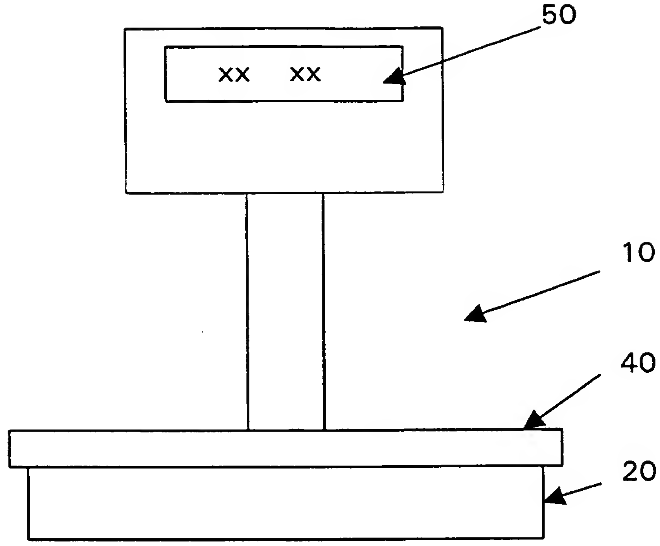
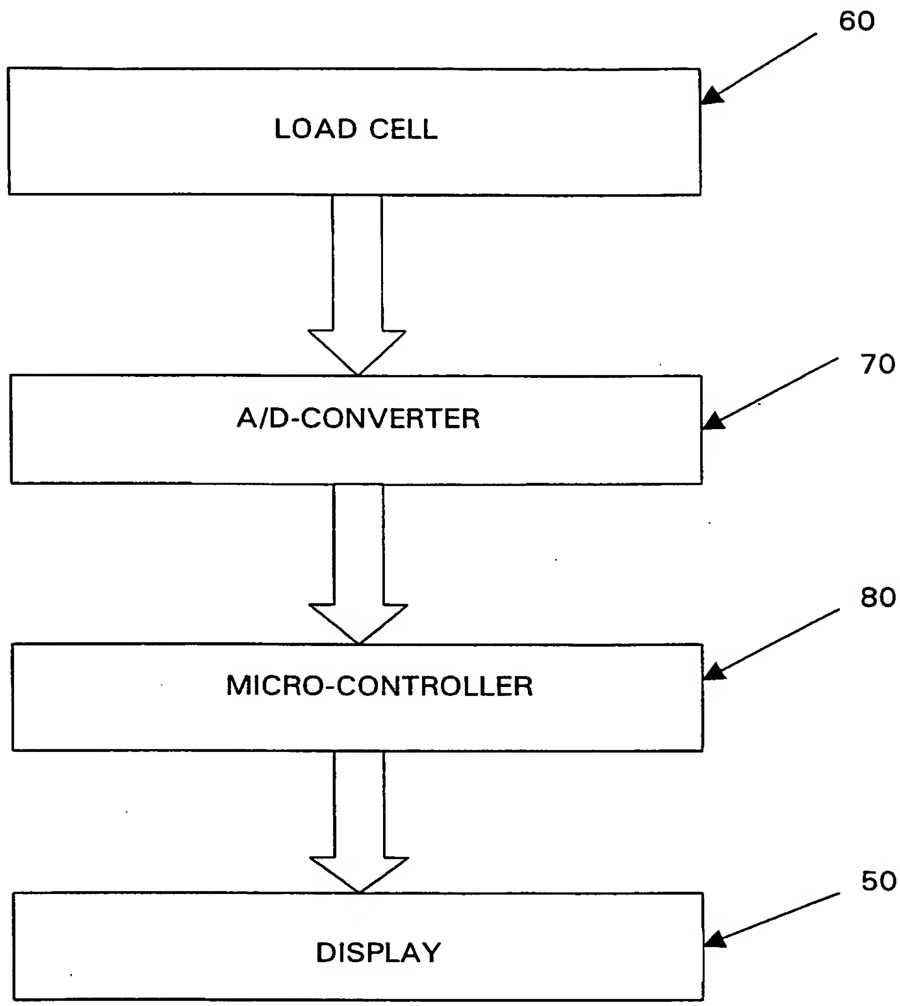


Fig. 1

5207 01.10.99



5207 01.12.99

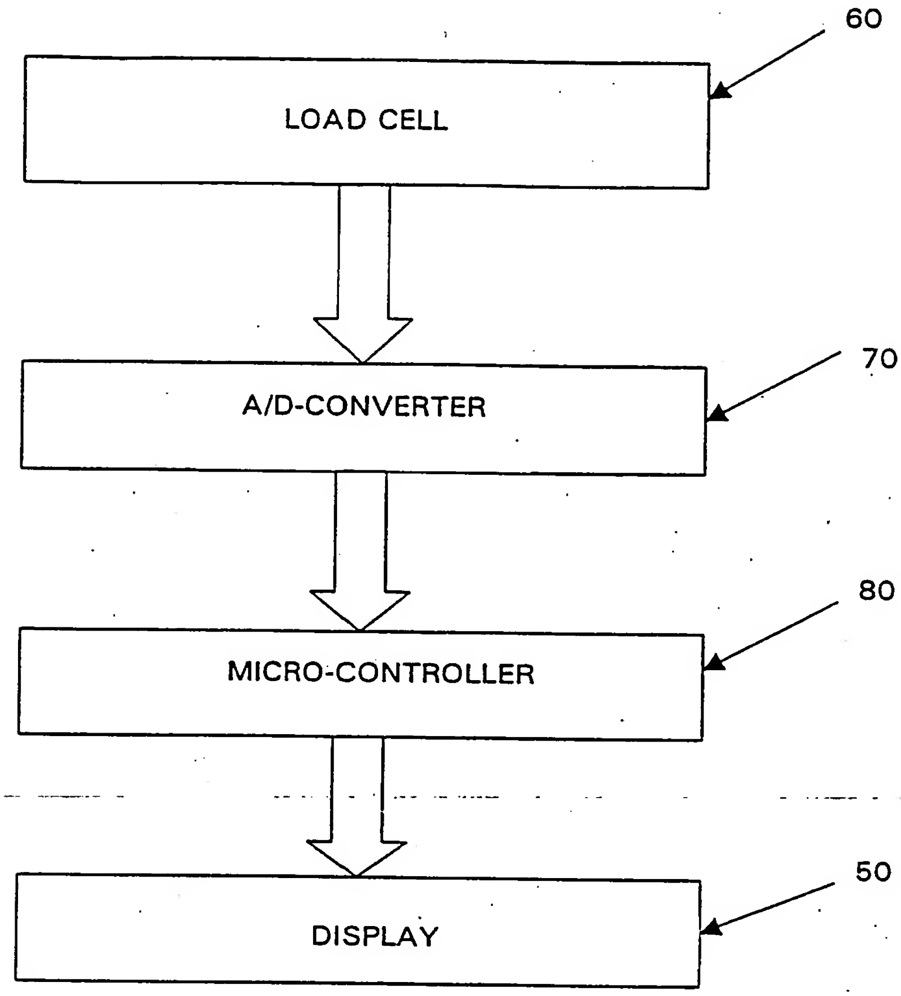


Figure 2

3 2 0 7 01.10.99

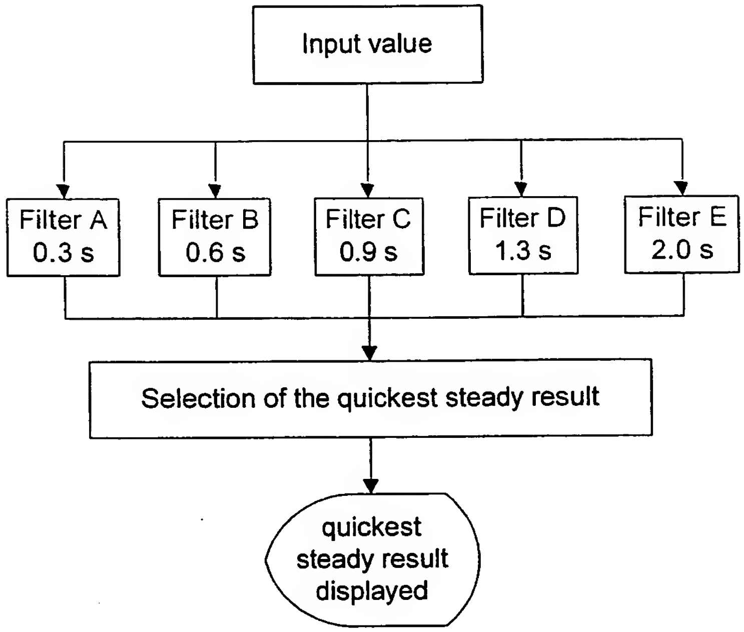


Figure 3

01.10.99

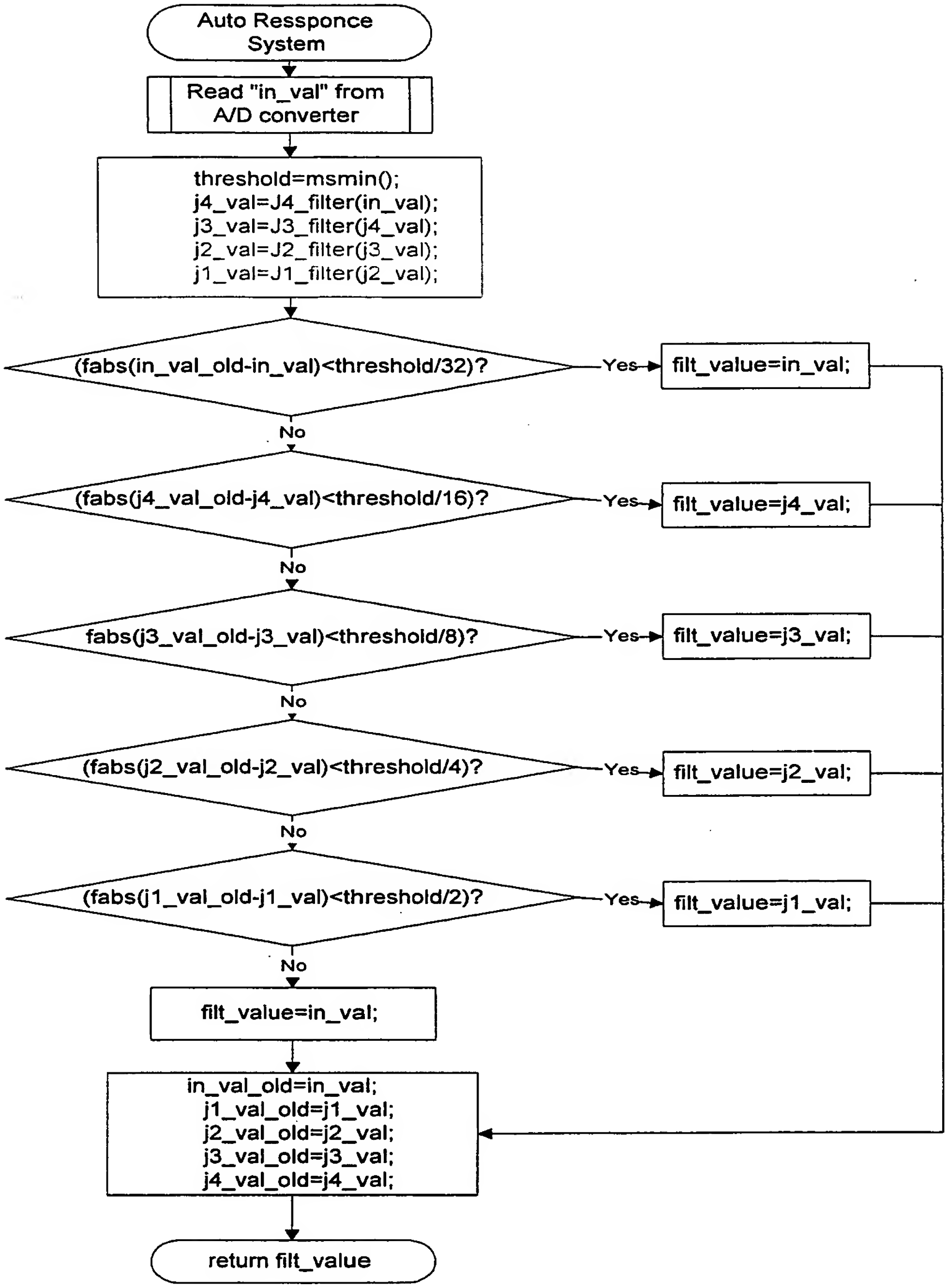


Figure 4.

```
// Beginning of the filter program:

#define J1_SIZE   16                    // max size of J1
#define J2_SIZE   8                     // max size of J2
#define J3_SIZE   4                     // max size of J3
#define J4_SIZE   2                     // max size of J4

static float  J1_array[J1_SIZE];
static float  J1_sum;                   // sum of values in filter array
static byte J1_pos;                     // position within filter array

static float  J2_array[J2_SIZE];
static float  J2_sum;                   // sum of values in filter array
static byte J2_pos;                     // position within filter array
static float  J3_array[J3_SIZE];
static float  J3_sum;                   // sum of values in filter array
static byte J3_pos;                     // position within filter array

static float  J4_array[J4_SIZE];
static float  J4_sum;                   // sum of values in filter array
static byte J4_pos;                     // position within filter array

// recalculate all sums in j-filters
void reset_J_filters(void)
{
 byte i;

 // reset J1_filter
 J1_sum=0;
 for (i=0;i<J1_SIZE;i++)  J1_sum += J1_array[i];

 // reset J2_filter
 J2_sum=0;
 for (i=0;i<J2_SIZE;i++)  J2_sum += J2_array[i];

 // reset J3_filter
 J3_sum=0;
 for (i=0;i<J3_SIZE;i++)  J3_sum += J3_array[i];

 // reset J4_filter
 J4_sum=0;
 for (i=0;i<J4_SIZE;i++)  J4_sum += J4_array[i];
}
```

```
// J1_filter -- return filtered value
float J1_filter(float value)
{
 J1_sum += value - J1_array[J1_pos];
 J1_array[J1_pos] = value;
 J1_pos = (J1_pos + 1) & (J1_SIZE-1);
 return J1_sum / J1_SIZE;
}

// J2_filter -- return filtered value
float J2_filter(float value)
{
 J2_sum += value - J2_array[J2_pos];
 J2_array[J2_pos] = value;
 J2_pos = (J2_pos + 1) & (J2_SIZE-1);
 return J2_sum / J2_SIZE;
}

// J3_filter -- return filtered value
float J3_filter(float value)
{
 J3_sum += value - J3_array[J3_pos];
 J3_array[J3_pos] = value;
 J3_pos = (J3_pos + 1) & (J3_SIZE-1);
 return J3_sum / J3_SIZE;
}

// J4_filter -- return filtered value
float J4_filter(float value)
{
 J4_sum += value - J4_array[J4_pos];
 J4_array[J4_pos] = value;
 J4_pos = (J4_pos + 1) & (J4_SIZE-1);
 return J4_sum / J4_SIZE;
}


float auto_filter(float in_val)
{
   MU8 filtnr;
   float j1_val,j2_val,j3_val,j4_val,threshold,filt_value;
   static float j1_val_old,j2_val_old,j3_val_old,j4_val_old,in_val_old;

   threshold=msmin();                          // get threshold value

   j4_val=J4_filter(in_val);
   j3_val=J3_filter(j4_val);
   j2_val=J2_filter(j3_val);
   j1_val=J1_filter(j2_val);
```

```
if (fabs(in_val_old-in_val)<threshold/32)
    {
        filt_value=in_val;
        filtnr=5;
    }
else if (fabs(j4_val_old-j4_val)<threshold/16)
    {
        filt_value=j4_val;
        filtnr=4;
    }
else if (fabs(j3_val_old-j3_val)<threshold/8)
    {
        filt_value=j3_val;
        filtnr=3;
    }
else if (fabs(j2_val_old-j2_val)<threshold/4)
    {
        filt_value=j2_val;
        filtnr=2;
    }
else if (fabs(j1_val_old-j1_val)<threshold/2)
    {
        filt_value=j1_val;
        filtnr=1;
    }
else
    {
        filt_value=in_val;
        filtnr=6;
    }

  in_val_old=in_val;
  j1_val_old=j1_val;
  j2_val_old=j2_val;
  j3_val_old=j3_val;
  j4_val_old=j4_val;

  //dbprintf("%d",filtnr);

  return filt_value;
}

// End of the filter program.
```

Fig. 5

ABSTRACT

The present invention relates to a method for weighing items and in particular to a method of filtering signals representing the weight of an item. The invention is applicable in all measuring instruments which indicate a DC value calculated from a signal which also has unwanted AC components of one or more unknown frequencies. This is, for example, the case in voltmeters, ampere meters and weighing-scales. Normally, the indicated result is the mean value of the signal measured for a much longer time than the expected period of the slowest AC component. This method is well known and reliable, but has the drawback of being very slow in situations where measuring speed is of importance.